An application of LDPC code for wireless coherent-light commutation in
atmospheric channel

Ning Hao\textsuperscript{a}, Yang’an Zhang \textsuperscript{a}, Jinnan Zhang \textsuperscript{a}, Minglun Zhang \textsuperscript{a}, Xueguang Yuan \textsuperscript{a}
\textsuperscript{a} State Key Laboratory of Information Photonics and Optical Communications of Beijing University of Posts and Telecommunications, P.O. Box 66 (BUPT)
Beijing, China 100876, Tel: 86-10-61198081
e-mail: haoning212@126.com

Abstract—Low Density Parity Check code is more and more taken seriously in high-speed transmission. In this article we represent a LDPC coder and decoder which based on IEEE802.16e and realize the coder and decoder with Virtex-5 FPGA. By using Matlab to make an off-line system simulation, we analyzed and compared the LDPC performance under the different length of code for LDPC coder then analyzed the influence of different iteration to the LDPC BER performance of decoder.

Keywords—Low Density Parity Check code(LDPC) , IEEE802.16e, Bit-Flipping algorithm (BF), parity check matrix

I. INTRODUCTION

LDPC is short for Low Density Parity Check codes which presented by Gallager at 1960s. This code is a kind of liner block code that based on parity check matrix. In the code there are a little non-zero elements and much more zero so is named. But the advantage of LDPC was ignored for a long time, for the reason that there were no effective methods to calculate in 1960s and also the Concatenated code was regarded as a better performance in coder at that time. While in 1996, Mac Kay found that the performance of LDPC has the ability of exceeding the Turbo code, so it became famous again and the researching of LDPC steps into a new period. Reports of the researching about LDPC indicate that performance of LDPC could approach to the Shanno limit in the last few years. In BIAWGN there is only 0.00d5dB difference from Shannon limit by using 1/2 code rate LDPC.

Bipartite graph could be used to indicate the LDPC, which is made up by Variable Nodes, Check Nodes and the Edges that between them. Variable Nodes indicate the information bit, so as to a column of parity check matrix. Check Nodes indicate the check sequence, so as to a row of parity check matrix. Then the Edges connect the Variables and Check Nodes, which indicate the non-zero elements in the parity check matrix. In bipartite graph, if a path from one node could loopback to itself is called ring. It is very harmful appearance when to decode a sequence which has a lot short rings in it. So it is better that avoiding rings when coding the sequence.

In this experiment a million gates FPGA of Xilinx Virtex-5 is used to realize the LDPC coder and LDPC decoder, the system block graph shown as below,
II. SYSTEM SETUP

A. Coder of LDPC module

For efficient encoding of LDPC, $H$ are divided into the form

$$H = \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}$$

Where $A$ is $(m - z) \times k$, $B$ is $(m - z) \times z$, $T$ is $(m - z) \times (m - z)$, $C$ is $z \times k$, $D$ is $z \times z$, and finally $E$ is $z \times (m - z)$. $B$ and $D$ correspond to the expanded $\tilde{h}_i$ and $\tilde{h}(m_i - 1)$, respectively.

Let $\nu = (u, p_1, p_2)$ where $u$ denotes the systematic part, $p_1$ and $p_2$ combined denote the parity part, $p_i$ has length $z$, and $p_2$ has length $(m - z)$. The definition equation $(H \cdot \nu^T) = 0$ splits into two equations, as in Equation (A.2) and Equation (A.3).

$$Au_i + Bp_i^T + Tp_i^T = 0 \quad (A.1)$$

and

$$(ET^{-1}A + C)u_i^T + (ET^{-1}B + D)p_i^T = 0 \quad (A.2)$$

Define $\phi = (ET^{-1}B + D)$ and with the parity check matrix as indicated $\phi = I$. Then from Equation (A.2), it can be concluded that

$$P_i^T = (ET^{-1}A + C)u_i^T \quad (A.3)$$

and

$$p_i^T = T^{-1}(Au_i + Bp_i^T) \quad (A.4)$$

As a result, the encoding procedures and the corresponding operation can be summarized below and illustrated in Figure 4.

1. Compute $Au_i$ and $Cu_i$.
2. Compute $ET^{-1}(Au_i)$.
3. Compute $p_i^T$ by $p_i^T = ET^{-1}(Au_i) + Cu_i^T$.
4. Compute $p_i^T$ by $Tp_i^T = Au_i^T + Bp_i^T$.

B. Decoder of LDPC module

Bit-Flipping, BF algorithm is a kind of hard decision decoding algorithm especially for LDPC which firstly proposed by Gallager. It is very easy to be realized on hardware and it has a very simple structure, because in the process of decoding only to handle the bit 0 and bit 1.

Suppose that there is an information sequence $\hat{\nu} = (a_0, a_1, \ldots, a_{m-1})$. Then coding this sequence to a sequence $\tilde{\nu} = (v_0, v_1, \ldots, v_{N-1})$ whose length is $N$. Sequence $\tilde{\nu} = (v_0, v_1, \ldots, v_{N-1})$ is used to transmitted in the channel and receiver could receive the information sequence $\tilde{\nu} = (y_0, y_1, \ldots, y_{N-1})$ and then it will make a hard decision to get a new sequence $\tilde{z} = (z_0, z_1, \ldots, z_{N-1})$ from $\tilde{\nu}$. The check matrix $H$ of LDPC is $H = [\tilde{h}_0, \tilde{h}_1, \ldots, \tilde{h}_{m-1}]^T$, which $\tilde{h}_i = (h_0, h_1, \ldots, h_{N-1})^T, M = N - K$.

Using hard decision sequence $\tilde{z}$ and check matrix $H$ to calculate syndrome vector $S$, shown as below,

$$S = (s_0, s_1, \ldots, s_{m-1})$$

$$= ZH^T \quad (B.1)$$

$$= (z_0, z_1, \ldots, z_{N-1}) \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0N-1} \\
               h_{10} & h_{11} & \cdots & h_{1N-1} \\
               \vdots & \vdots & \ddots & \vdots \\
               h_{M-10} & h_{M-11} & \cdots & h_{M-1N-1} \end{bmatrix}$$

Which $s_i$ is,

$$s_i = Z\tilde{h} = \sum_{j=0}^{M-1} z_{j} h_{ij} \quad (B.2)$$

The addition functions above are Mod-$2$ add operation. Vector $\tilde{S}$ indicates whether the function is satisfied. If $\tilde{S}_i$ equals to $0$, then sequence $Z$ is satisfied with the constraints relation of row $i$ in check matrix. If vector $\tilde{S}$ equals to $0$, vector, means that all elements are satisfied with the check matrix $H$ and then finally indicate that it got a exactly correct decoding sequence. Else there are non-zero elements in vector $\tilde{S}$ indicates that there are some error bit in $Z$ sequence and still should do more other processing. Using the vector $S$ and check matrix $H$ to calculate a new syndrome sequence $F$, then choosing the most unsatisfied function to reverse the relevant bit or bits.

$$\tilde{F} = (f_0, f_1, \ldots, f_{M-1})$$

$$= SH \quad (B.3)$$

$$= (s_0, s_1, \ldots, s_{m-1}) \begin{bmatrix} h_{00} & h_{01} & \cdots & h_{0N-1} \\
               h_{10} & h_{11} & \cdots & h_{1N-1} \\
               \vdots & \vdots & \ddots & \vdots \\
               h_{M-10} & h_{M-11} & \cdots & h_{M-1N-1} \end{bmatrix}$$

$$f_i = \sum_{j=0}^{M-1} s_j h_{ij} \quad (B.4)$$

And here are the procedure of BF algorithm,
(1) make a hard decision of input sequence, if $y_k > 0$ then $Z^0_k = 1$, else $Z^0_k = 0$, so it will return out the initial decoding sequence $\tilde{Z} = (z^0_1, ..., z^0_N)$. The iterations is set to $k=1$;
(2) multiplying $Z^k$ with check matrix $H^T$, then result is in the vector $\tilde{S} = (s^k_1, s^k_2, ..., s^k_N)$;
(3) counting the numbers of each node in $\tilde{v}_k$ which is not equals to the check function $f^k_i = \sum_{j} S^k_{i,j}$;
(4) If vector not zero vector $f^k_i = \sum_{j} S^k_{i,j}$ then multiplying $S^k_i$ with check matrix $H$, then result in the vector $\tilde{F}^k_i = (f^k_1, f^k_2, ..., f^k_N)$;
(5) Finding the biggest element in vector $\tilde{F}^k_i = (f^k_1, f^k_2, ..., f^k_N)$ and revert the information bit that the biggest elements corresponded then result is a new sequence;
(6) Repeat the step 2 to step 5 until the elements in vector $S$ are all satisfied with the check matrix $H$ or the iterations gets the top value that system set.

III. RESULT AND ANALYSIS

By using Matlab to simulate LDPC of different length of 576, 1152 and 2304 at the condition of same code rate and 15 times of iteration. Then analyzing the influence of different length of LDPC to the BER performance. The result of simulation is shown as below,

From the Figure 6, it shows that the performance of LDPC is continuous improving with increasing of code length under the same SNB. In the low SNB area, it has little improvement to the BER when increase the length of code. But with the increasing of SNB, the BER of LDPC has been obviously reduced. At the BER of, compared the code length of 1152 and 576 LDPC with 2304, the SNB has been decreased about 0.4dB and 0.8 dB. While with increasing of length, the improvement of performance is relative. When the length of code gets a relative value, the improvement of performance no longer increasing. For the reason that with the increasing of code length the complexity of coding and decoding also increased, and the LDPC performance will approach the limitation.

By using Matlab to analysis the 576 length, code rate of 3/4A LDPC with the different iteration of 5, 15 and 30. The result of simulation from Matlab in different iteration of LDPC shown as below,

From the Figure 7 it indicates that under the same SNB the performance of LDPC improved with the increasing of iteration times. At the BER of $10^{-5}$, compared the iteration of 15 and 5 times with 30 times, the SNB has been decreased about 0.2dB and 0.5 dB respectively. But the BER of LDPC couldn’t keep decreasing unlimitedly with the increasing of iteration times. No matter how to increase the iteration the performance of LDPC couldn’t improve, when the BER has decreased to a limited level. At this time the increasing of iteration only make the coding or decoding much more complicated.

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (2009RC0401), the China Postdoctoral Science Foundation funded project (20100470259). The fund of young scholar innovation project. (2011PTB-00-31). The fund of young scholar innovation project. (2012RC0407). The fund of young scholar innovation project. (2012RC0406).
REFERENCES


